1 31. (Amended) A graphics system for [processing parameter values of graphics
2 primitives in a] storing display list values in a register file, wherein the display list is
3 shortened to enable fast processing time while maintaining the quality of information
4 contained in the display list, the graphics system comprising:
5 a register file, for storing at least one set of parameter values, the register file
6 comprising a plurality of registers;
7 a load instruction unit, for storing an instruction having an opcode portion that
8 specifies a rendering operation, and a write-enable portion that spans a
9 plurality of bits, wherein a first bit corresponds to a target starting register file
10 address and subsequent bits sequentially correspond to register file addresses
11 that follow the target starting register file address;
12 a shifter coupled to receive the write-enable portion, [for] wherein the shifter
13 sequentially [performing] performs single-bit shifts upon the contents of the
14 write-enable portion; and
15 a rendering parameter storage controller coupled to the shifter and the register file,
16 [for sequentially stepping] wherein the rendering parameter storage controller
17 sequentially steps through register file addresses [corresponding to bits
18 spanning the] as write-enable portion bits are considered by the shifter, and
19 [storing a parameter] stores a display list value in the register file in response
20 to a write-enable portion bit under consideration by the shifter having a
21 predetermined value.

26

33. (Amended) [A] In a computer system having a graphics controller, a method for storing [graphics primitive parameter values forming a] display list values in a register file comprising a plurality of addressable registers, wherein the display list is shortened to enable fast processing time while maintaining the quality of information contained in the display list, each display list value corresponding to a graphics primitive, the method comprising the steps of:

retrieving an instruction that includes an opcode portion and a write-enable portion, the opcode portion specifying a rendering operation, the write-enable portion spanning a plurality of bits, wherein a first bit corresponds to a target starting register file address, and subsequent bits sequentially correspond to register file addresses that follow the target starting register file address;

sequentially examining [bits] each bit within the write-enable portion; and

storing a [parameter] display list value in the register file in response to a write-enable portion bit under examination having a predetermined value.

REMARKS

Claims 10 and 31-34 were filed in this case on October 5, 1999, and all claims were rejected. Claim 10 has been canceled. Claims 31 and 33 have been amended. In view of the above changes and the following remarks, reconsideration of the application is respectfully requested.